CSCI–4190 Introduction to Robotic Algorithms, Spring 2006

Writing standards-compliant C++

We compile and test your programs in Linux (using g++ 3.x) because it's what we use on our computers. Since many of you use MSVC++ for development, there are often a number of issues in getting your code to compile since VC++ is frequently standards-deficient.

In grading your Assignment 1 submissions, Kris encountered a number of problems while compiling your code. For future assignments, we will offer a 5 point bonus if your code compiles without modification. Here are some tips for writing programs that will work with compilers other than VC++.

- Use `fabs` for absolute values; on most compilers `abs` only operates on integers so whatever you pass it will be truncated.
- Variable scoping: you should not do the following:

  ```
  for(int i = 0; i < something; ++i) { do stuff }
  if(i == x) { do stuff }
  ```

  In other words, a variable declared during `for` loop initialization goes out of scope after the `for` loop exits in ISO C++, so this code won't compile using, e.g., g++.
- Don't use `itoa`: such a function doesn't exist in most libc implementations.
- If you need π, use the following:

  ```
  #include <math.h>
  #ifndef M_PI
    #define M_PI 3.14159265358979323846264338327
  #endif
  ```

  Most sane `math.h` implementations define `M_PI`, but some (MSVC++) do not. Either way, you probably shouldn't hardcode "3.14" throughout your code since in many cases it's not nearly accurate enough.
- On any operating system other than windows or with any compiler other than MSVC++, `#includes` are case sensitive:
  - **BAD**: `#include <Math.h>`
  - **GOOD**: `#include <math.h>`
- There's a function called `random` in most libc implementations, so don't write your own function called that.
- If you're sorting `vector<YourType>`, `YourType` must have an `operator<` member, *and* it must be of the form:

  ```
  bool operator<(const YourType &y) const
  ```

  Many STL implementations require the `const` declarations.